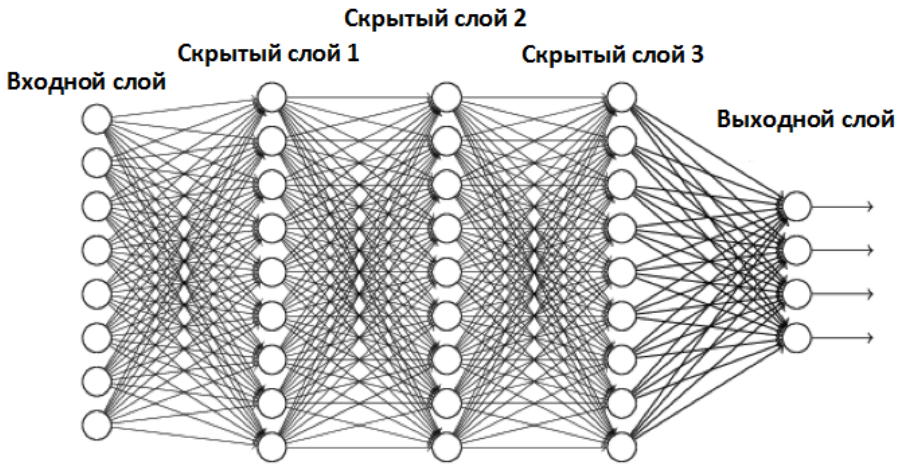
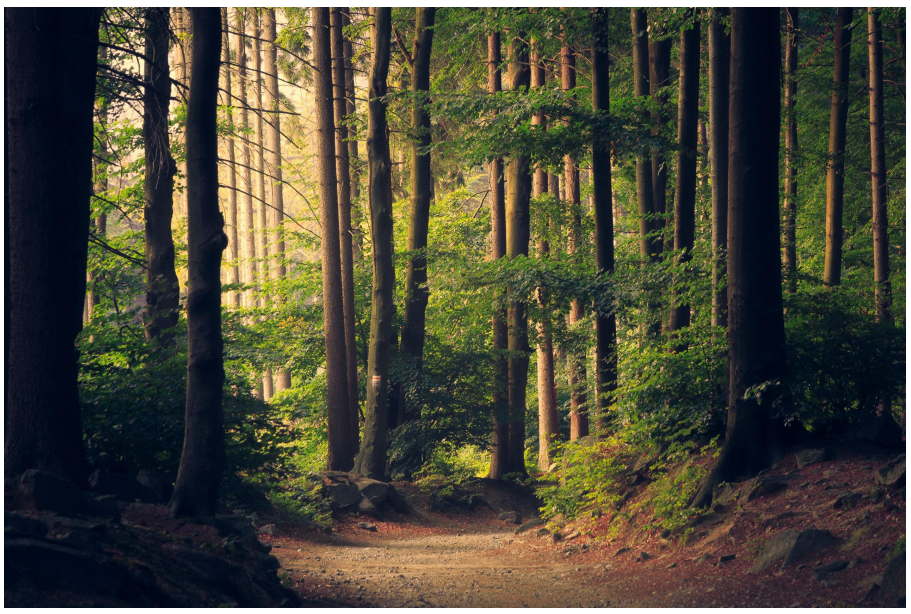


Объяснение деревьев решений Scikit-Learn

Дата публикации Feb 23, 2019



фотоЛукаш ШмигельнаUnsplash

Деревья решений являются наиболее важными элементами случайного леса. Они способны подгонять сложные наборы данных, позволяя пользователю увидеть, как было принято решение. При поиске в Интернете я не смог найти ни одной понятной статьи, которая могла бы легко их описать, поэтому здесь я пишу о том, что я узнал до сих пор. Важно отметить, что одно дерево решений не очень хороший предсказатель; однако, создавая их совокупность (лес) и собирая их прогнозы, можно получить один из самых мощных инструментов машинного обучения - так называемый случайный лес.

Make sure you have installed pandas and scikit-learn on your machine. If you haven't, you can learn how to do so here.

Дерево решений Scikit-Learn

Давайте начнем с создания дерева решений, используянабор данных радужной оболочкит. Набор данных радужной оболочки содержит четыре объекта, три класса цветов и 150 образцов.

Особенности:длина чашелистика (см), ширина чашелистика (см), длина лепестка (см), ширина лепестка (см)

Классы:сетоза, версиколор, вирджиния

Численно, цветы сетозы идентифицируются по нулю, разноцветные - по одному, а virginica - по два.

Для простоты мы обучим наше дерево решений, используя все функции и установив глубину в два.

```
# Importing Libraries
import pandas as pd
from sklearn.datasets import load_iris
```

```python
from sklearn.tree import DecisionTreeClassifier

# Load data and store it into pandas DataFrame objects
iris = load_iris()
X = pd.DataFrame(iris.data[:, :], columns = iris.feature_names[:])
y = pd.DataFrame(iris.target, columns =["Species"])

# Defining and fitting a DecisionTreeClassifier instance
tree = DecisionTreeClassifier(max_depth = 2)
tree.fit(X,y)
```

CreateDecisionTree.py hosted with ❤ by GitHub view raw

Визуализация дерева решений

Конечно, мы до сих пор не знаем, как это дерево классифицирует образцы, поэтому давайте визуализируем это дерево, сначала создав файл точек с помощью Scikit-Learn.export_graphvizмодуль, а затем обрабатывает егоGraphviz,

```python
# Visualize Decision Tree
from sklearn.tree import export_graphviz

# Creates dot file named tree.dot
export_graphviz(
            tree,
            out_file =  "myTreeName.dot",
            feature_names = list(X.columns),
            class_names = iris.target_names,
            filled = True,
            rounded = True)
```

VisualizeDecisionTree.py hosted with ❤ by GitHub view raw

Это создаст файл с именем tree.dot, который должен быть обработан на*Graphviz*,ВотУчебник YouTubeэто показывает вам, как обрабатывать такой файл с помощью GraphViz. Конечный результат должен быть аналогичен показанному в**Рисунок 1**; однако, может появиться другое дерево, даже если тренировочные данные одинаковы!

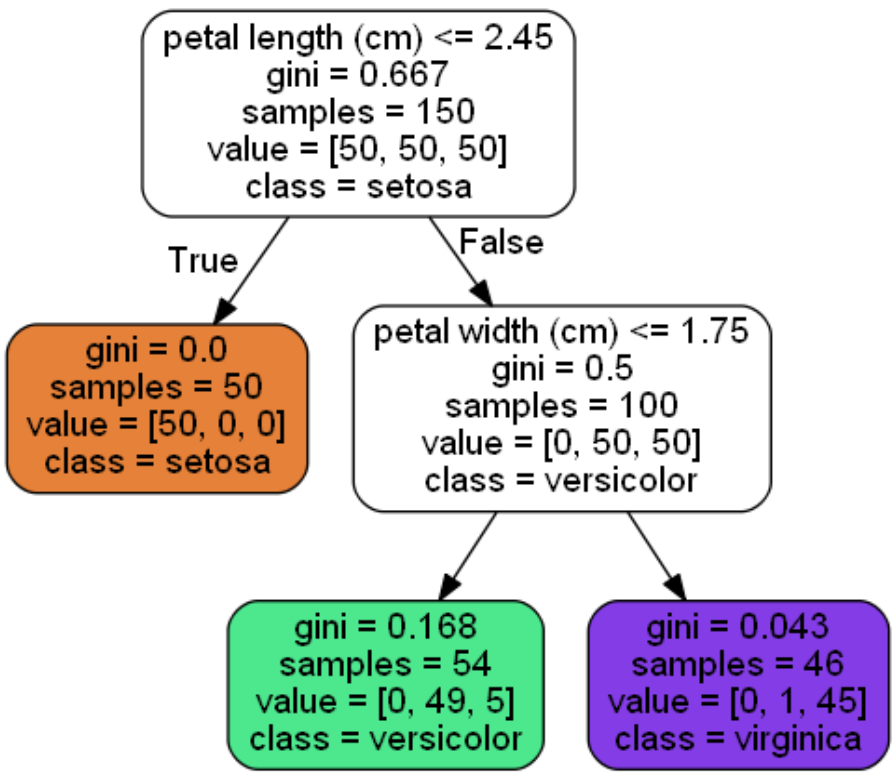


Рисунок-1) Наше дерево решений:В этом случае узлы окрашены в белый цвет, а листья окрашены в оранжевый, зеленый и фиолетовый. Подробнее о листьях и узлах позже.

Единое дерево решений является классическим примером типа классификатора, известного как*белая коробка*, Предсказания, сделанные классификатором белого ящика, могут быть легко поняты. Вот отличныйстатьяо черно-белой коробке классификаторов.

Понимание содержания узла

В**Рисунок 1**Вы можете видеть, что каждая коробка содержит несколько характеристик. Давайте начнем с описания содержимого самого верхнего узла, чаще всего называемого*корневой узел*, Корневой узел находится на глубине нуля, см.**Фигура 2**, Узел - это точка в дереве решений, где задается вопрос. Это действие делит данные на меньшие подмножества.

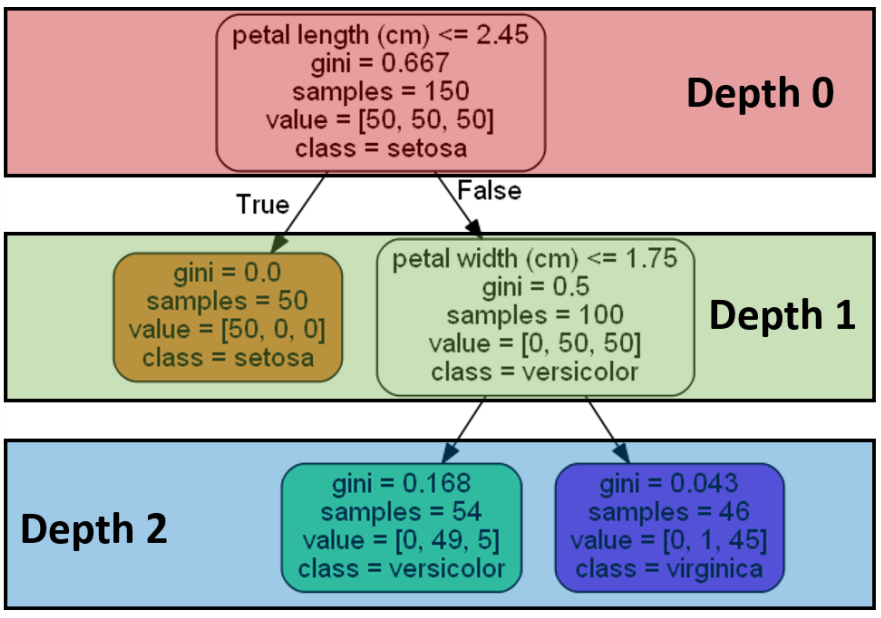


Рисунок-2) Глубина дерева:Светлые коробки иллюстрируют глубину дерева. Корневой узел расположен на глубине нуля.

длина лепестка (см) <= 2,45:Первый вопрос, который задает дерево решений, - это если длина лепестка меньше 2,45. Основываясь на результате, он следует либо по истинному, либо по ложному пути.

Джини = 0,667:Оценка Джини - это показатель, который количественно определяет чистоту узла / листа (подробнее о листьях чуть-чуть). Оценка Джини больше нуля означает, что образцы, содержащиеся в этом узле, относятся к разным классам. Оценка Джини, равная нулю, означает, что узел чистый, что в этом узле существует только один класс выборок. Вы можете узнать больше о мерах загрязненияВот, Обратите внимание, что у нас показатель Джини больше нуля; поэтому мы знаем, что образцы, содержащиеся в корневом узле, относятся к разным классам.

образцы = 150:Поскольку набор данных цветов радужной оболочки содержит 150 образцов, это значение установлено равным 150.

значение = [50, 50, 50]:`value`Список говорит вам, сколько образцов в данном узле попадают в каждую категорию. Первый элемент списка показывает количество образцов, принадлежащих классу setosa, второй элемент списка показывает количество образцов, принадлежащих классу versicolor, а третий элемент списка показывает количество принадлежащих ему образцов. в классе Вирджиники. Обратите внимание, что этот узел не является чистым, поскольку в одном и том же узле содержатся разные типы классов. Мы знали это уже по счету Джини, но на самом деле приятно видеть это.

класс = сетоса:`class`значение показывает прогноз, который сделает данный узел, и его можно определить по`value`список. Какой бы класс ни встречался в узле, он будет выбран как`class`ценность. Если бы дерево решений заканчивалось в корневом узле, было бы предсказано, что все 150 выборок принадлежали к классу setosa. Конечно, это не имеет смысла, поскольку для каждого класса существует одинаковое количество выборок. Мне кажется, что дерево решений запрограммировано на выбор первого класса в списке, если для каждого класса имеется равное количество выборок.

Понимание того, как дерево делает раскол

Чтобы определить, какую функцию использовать для первого разделения, то есть для создания корневого узла, алгоритм выбирает функцию и выполняет разбиение. Затем он смотрит на подмножества и измеряет их примеси с использованием показателя Джини. Он делает это для нескольких порогов и определяет, что наилучшее разделение для данной функции - это то, которое создает самые чистые подмножества. Это повторяется для всех функций в обучающем наборе. В конечном счете, корневой узел определяется функцией, которая производит разделение с наиболее чистыми подмножествами. Как только корневой узел определен, дерево растет до глубины одного. Тот же процесс повторяется для других узлов дерева.

Понимание того, как дерево сделает прогноз

Предположим, у нас есть цветок с`petal_length = 1`а также`petal_width = 3`, Если мы будем следовать логике дерева решений, показанного на**Рисунок 1**, мы увидим, что мы окажемся в оранжевой коробке. В**Рисунок 1**Если вопрос, который задает узел, оказывается верным (ложным), мы переместимся влево (вправо). Оранжевая коробка на глубине одного, см.**Фигура 2**, Поскольку из этой коробки ничего не растет, мы будем называть ее*листовой узел*, Обратите внимание на сходство этого фактического дерева, см.**Рис-3**, Кроме того, обратите внимание, что показатель Джини равен нулю, что делает его чистым листом. Общее количество образцов составляет 50. Из 50 образцов, которые заканчиваются на узле оранжевого листа, мы можем видеть, что все они принадлежат к классу setosa, см.`value` список для этого листа. Следовательно, дерево будет предсказывать, что образец представляет собой цветок сетузы.

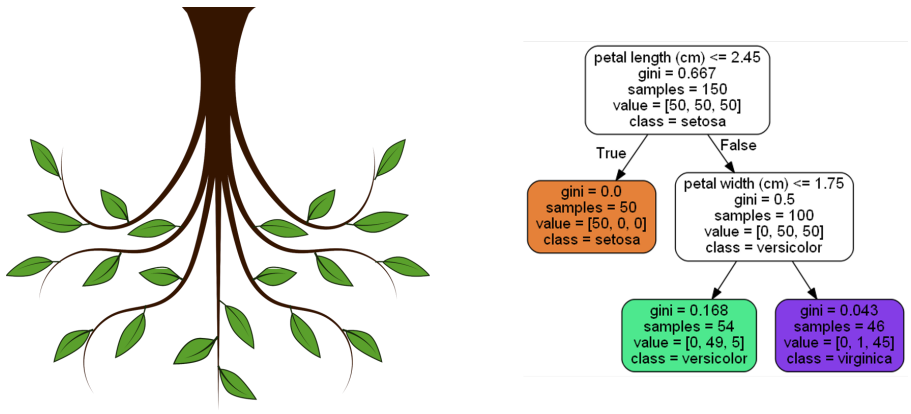


Рисунок-3) Реальное дерево против дерева решенийДерево слева перевернуто, чтобы проиллюстрировать, как дерево растет от своего корня и заканчивается на его листьях. Видение дерева решений справа должно сделать эту аналогию более ясной.

Давайте выберем более интересный образец. Например,`petal_length = 2.60`а также`petal_width = 1.2`, Мы начинаем с корневого узла, который спрашивает, меньше ли длина лепестка 2,45. Это ложно; Поэтому мы переходим к*внутренний узел*справа, где показатель Джини равен 0,5, а общее количество выборок равно 100. Этот внутренний узел на глубине 1 задаст вопрос:*Ширина лепестка меньше 1,75*? »В нашем случае это действительно так, поэтому мы переместимся влево и окажемся в узле листа зеленого цвета, который находится на глубине 2. Дерево решений предскажет, что этот образец является многоцветным цветком. Вы можете видеть, что это наиболее вероятный случай, потому что 49 из 54 образцов, которые заканчиваются в узле зеленого листа, были разноцветными цветами, см.`value`список для этого листа.

Создание прогноза для новых образцов с использованием обученного дерева

Теперь, когда мы знаем, как работает наше дерево решений, давайте делать прогнозы. Ввод должен быть в списке и упорядочен как`[sepal length, sepal width, petal length, petal width]`где длина чашелистика и ширина чашелистика не влияют на прогнозы, сделанные деревом решений, показанным в**Рисунок 1**; поэтому мы можем присвоить им произвольное значение.

```
# Making a Prediction On a New Sample
sample_one_pred = int(tree.predict([[5, 5, 1, 3]]))
sample_two_pred = int(tree.predict([[5, 5, 2.6, 1.5]]))
print(f"The first sample most likely belongs a {iris.target_names[sample_one_pred]} flower.")
print(f"The second sample most likely belongs a {iris.target_names[sample_two_pred]} flower.")
```

makingDecionTreePrediction.py hosted with ❤ by GitHub view raw

Выход должен быть:

```
The first sample most likely belongs a setosa flower.
The second sample most likely belongs a versicolor flower.
```

Это именно то, что мы предсказали, следуя логике дерева решений.

Параметры дерева решений Scikit-Learn

Если вы посмотрите напараметры DecisionTreeClassifierможет взять, вы можете быть удивлены, давайте посмотрим на некоторые из них.

критерий: Этот параметр определяет, как будет измеряться примесь. Значением по умолчанию является «Джини», но вы также можете использовать «энтропию» в качестве метрики для примесей.

разветвитель:Вот как дерево решений ищет функции для разделения. Значение по умолчанию установлено на «лучший». То есть для каждого узла алгоритм учитывает все функции и выбирает наилучшее разбиение. Если вы решите установить параметр разделителя на «случайный», то будет рассматриваться случайное подмножество объектов. Разделение будет затем выполнено наилучшей функцией в случайном подмножестве. Размер случайного подмножества определяется параметром max_features. Это частично, где Случайный Лес получает свое имя.

Максимальная глубина:Это определяет максимальную глубину дерева. В нашем случае мы используем глубину два, чтобы сделать наше дерево решений. По умолчанию установлено значение none. Это часто приводит к переопределенным деревьям решений. Параметр глубины является одним из способов, которыми мы можем упорядочить дерево или ограничить его рост, чтобы предотвратить**над-фитинга**, В**Рисунок-4**, вы можете увидеть, что произойдет, если вы не установите глубину дерева - чистое безумие!

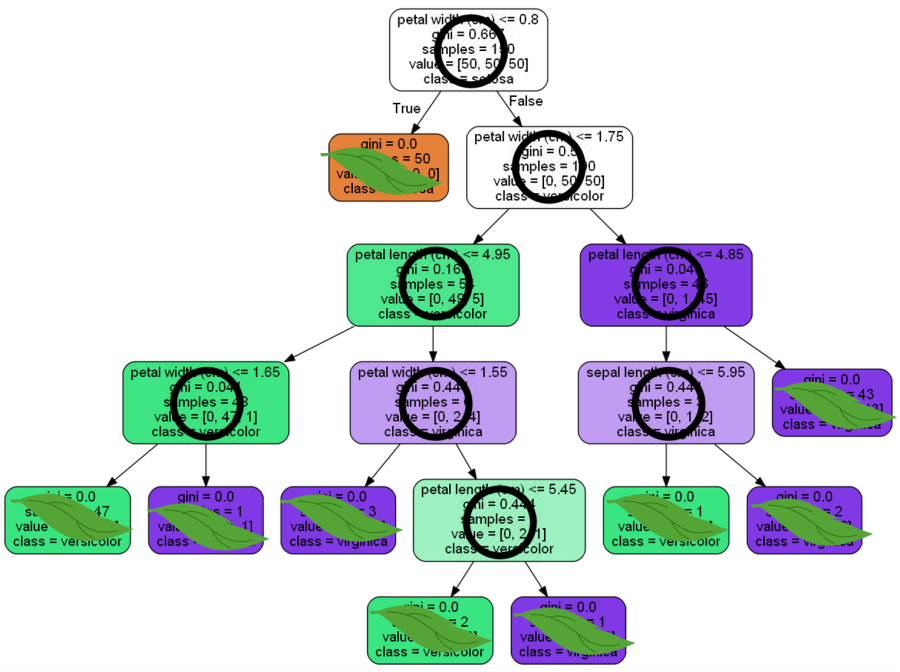


Рисунок-4) Полностью выросшее дерево решений:В дереве, показанном выше, ни один из параметров не был установлен. Дерево вырастает до глубины пяти. Есть восемь узлов и девять листьев. Не ограничение роста дерева решений может привести к чрезмерной подгонке.

min_samples_split:Минимальное количество выборок, которое должен содержать узел, чтобы рассмотреть расщепление. Значение по умолчанию равно двум. Вы можете использовать этот параметр, чтобы упорядочить ваше дерево.

min_samples_leaf:Минимальное количество выборок необходимо считать листовым узлом. Значение по умолчанию установлено в единицу. Используйте этот параметр, чтобы ограничить рост дерева.

max_features:Количество функций, которые следует учитывать при поиске лучшего разделения. Если это значение не установлено, дерево решений будет учитывать все функции, доступные для наилучшего разделения В зависимости от вашего приложения, часто бывает полезно настроить этот параметр.Вот статья, которая рекомендует, как установить max_features.

В целях синтаксиса, давайте установим некоторые из этих параметров:

```
# Setting parameters
tree = DecisionTreeClassifier(criterion = "entropy", splitter = "random", max_depth = 2,  min_samples_split =
                              min_samples_leaf = 2, max_features = 2)
tree.fit(X,y)
```

DecisionTreeSetParameters.py hosted with ❤ by GitHub view raw

Заключительные замечания

Теперь вы знаете, как создать дерево решений с помощью Scikit-learn. Что еще более важно, вы должны иметь возможность визуализировать его и понимать, как он классифицирует образцы. Важно отметить, что нужно ограничить свободу дерева решений. Есть несколько параметров, которые могут упорядочить дерево. По умолчанию для max_depth установлено значение none. Таким образом, дерево будет расти полностью, что часто приводит к переоснащению. Более того, единое дерево решений не является очень мощным предиктором.

Реальная сила деревьев решений раскрывается в большей степени при культивировании многих из них - при ограничении их роста - и при сборе их индивидуальных предсказаний для формирования окончательного заключения. Другими словами, вы выращиваете лес, и если ваш лес носит случайный характер, используя концепцию`splitter = "random"`Мы называем это Случайный Лес. Многие параметры, используемые в Scikit-Learn Random Forest, аналогичны тем, которые описаны в этой статье. Поэтому лучше понять, что такое единое дерево решений и как оно работает, прежде чем переходить к использованию больших пушек.

Вы можете найти меня вLinkedIn,

Оригинальная статья

- Фреймворки и библиотеки (большая подборка ссылок для разных языков программирования)
- Список бесплатных книг по машинному обучению, доступных для скачивания
- Список блогов и информационных бюллетеней по науке о данных и машинному обучению
- Список (в основном) бесплатных курсов машинного обучения, доступных в Интернете